

# frivol: Toteutusdokumentti

Topi Talvitie

16. kesäkuuta 2013

Projektissa toteutettiin Voronoi-diagrammin laskemiseen kirjasto frivol, sen visualisointiin JavaScript-ohjelma frivoldraw ja testit Boostin testiframeworilla.

## Aikavaativuus

Koko kirjaston ulkoinen rajapinta on frivol/frivol.hpp:ssa määriteltävä funktio, joka laskee annetusta  $n$  pisteestä Voronoi-diagrammin. Kirjastossa on mahdollista käyttää erilaisia tietorakenteita tapahtumaprioriteettijonoon ja beach linen hakupuuhun. Tässä aikavaativuusanalyysissä keskitytään oletustoteutuksiin, jotka ovat binäärikeko ja AVL-puu.

Fortunen algoritmi muodostuu tapahtumakäsittelyistä, jotka säilytetään prioriteettijonossa. Tapahtumia on yhteensä  $O(n)$  kappaletta, joten binäärikeolla tapahtumien hallintaan käytetään yhteensä  $O(n \log n)$  aikaa.

Jokaisessa tapahtumassa tehdään  $O(1)$  kyselyä/lisäystä/poistoa/siirtymistä beach linellä ja muutenkin  $O(1)$  operaatioita, joten beach linellä on  $O(n)$  kaarta ja koska kaikki operaatiot AVL-puussa ovat  $O(\log n)$ -aikaisia menee itse tapahtumien käsittelyyn korkeintaan  $O(n \log n)$  aikaa.

Siis koko algoritmin suoritus vie  $O(n \log n)$  aikaa.

Algoritmissa ainoat muistissa olevat asiat ovat tapahtumajono ( $O(n)$  tilaa), beach line ( $O(n)$  tilaa) ja tulosterakenne ( $O(n)$  tilaa, koska tunnetusti Voronoi-diagrammin koko on lineaarinen syötepisteiden lukumäärään). Siis algoritmin tilavaativuus on  $O(n)$ .

## Visualisointiohjelma frivoldraw

Jotta kirjaston tuottamia Voronoi-diagrammeja voitaisiin piirtää, toteutettiin visualisointia varten JavaScript-ohjelma frivoldraw. Koska kirjasto on

C++-kirjasto, käytettiin sen kääntämiseen emscripten-kääntäjää. Itse visualisaation käyttöliittymä toteutettiin JavaScriptillä HTML5 Canvasta käyttäen, ja kirjaston käyttämiseen JavaScriptistä toteutettiin pieni C-rajapinta `frivoldraw_lib`.

## Puutteet

frivoldrawilla Voronoi-diagrammeja piirrellessä kävi ilmi, ettei kirjasto vielä ole numeerisesti stabiili. Pisteitä ympäriinsä raahaillessa kävi välillä niin, että joidenkin pisteiden välinen Voronoi-särmä katosi ja kuvaan ilmestyi ylimääräisiä pitkiä särmiä. Tässä vaiheessa frivoldrawin kirjastolle syöttämät pisteet olivat kokonaislukukoordinaattisia, joten on syytä epäillä että ongelmakohdat liittyvät tapauksiin, joissa kolme pistettä on samalla suoralla - mikä on ymmärrettävää sillä niissä tapauksissa on vaikeaa päättää mikä on esimerkiksi pisteiden ympäripiirretyn ympyrän keskipiste (jolla Voronoi-solmun pitäisi sijaita). Kun frivoldrawiin lisättiin perturbaatio joka lisäsi jokaiseen koordinaattiin satunnaisen luvun väliltä  $[0, 1[$ , ongelmia ei enää ilmennyt, joten algoritmi kuitenkin vaikuttaisi toimivan melko tiheässä joukossa tapauksia.

Kuten helposti laskennallisen geometrian algoritmeissa käy, frivol ei toimi oikein aivan kaikissa tapauksissa, vaikka toimiikin jo erittäin varmasti perturboidulle datalle. Kirjaston korjaamiseen numeerisesti stabiiliksi ei ole enää aikaa, joten se jää myöhemmin parannettavaksi asiaksi.